Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A system for parsing a web-document based on elements, which is applied to an application of a handheld terminal and calls the web-document to provide it to the handheld terminal, comprising:

a word parser for separating a token on the basis of markup and non-markup by referring to a token table for all markup data necessary for kind of document to be supported; and

a syntax parser for parsing a contents model on the basis of document type definition (DTD) of each document, parsing each syntax on the basis of the result of parsing the contents model, and generating a tree-based object on the basis of graphic user interface (GUI) of the terminal.

2. (Original) The system of claim 1, wherein the word parser comprises:

a comment parser for processing a comment and a space;

a markup start parser for recognizing a markup start tag and generating a token;

an attribute parser for parsing an attribute and generating a token; and

a parsed character data analyzer for analyzing parsed character data and generating a token.

3. (Original) The system of claim 1, wherein the syntax parser comprises:

an XML verifier for verifying whether a corresponding document is composed suitable for each DTD on the basis of the token generated by the word parser; and

a terminal GUI-based object generator for matching the analyzed markup and a GUI of the terminal.

4. (Currently Amended) The system of ~~any one of claims 1 through 3~~claim 1, wherein the parsing system integrally parses a web-document composed on the basis of any one of SGML and XML related to HTML, XHTML, mHTML, cHTML, WML and HDML.

5. (Currently Amended) The system of ~~any one of claims 1 through 3~~claim 1, wherein the parsing system can be applied to any handheld terminal and select kind of an element to be parsed according to specification of each of the terminals.

6. (Original) A method for parsing a called web-document of a web-server, the method comprising the steps of:

(a) reading a token from the web-document and parsing the token;

(b) if the token is not a defined start tag or if the token is a comment or a space as result of the step (a), ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

(c) parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

(d) as the result of the step (c), if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

(e) in case the contents of the element are not the parsed character data, if the end tag corresponding to the start tag defined appears, terminating, and if the end tag does not appear, ignoring and returning.

7. (Original) The method of claim 6, wherein the step (c) comprises the steps of:

if the read token does not include a defined start tag, reading the data continuously until the end tag appears, thereby ignoring the token; and

reading a new token.

8. (Original) A recording medium for storing a program for parsing a called web-document of a web-server, the recording medium being read by a computer, the program comprising the functions of:

(a) reading a token from the web-document and parsing the token;

(b) if the token is not a defined start tag or if the token is a comment or a space as result of the function (a), ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

(c) parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

(d) if the contents of the element are parsed character data as result of the function (c), storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

(e) in case the contents of the element are not the parsed character data, if the end tag corresponding to the start tag defined appears, terminating, and if the end tag does not appear, ignoring and returning.

9. (Original) A system for parsing a web-document based on elements, which calls the web-document to provide it to a handheld terminal, comprising:

a word parser for extracting and separating all tokens of the web-document supplied regardless of kind of a markup language used to compose the web-document by referring to a token table; and

a syntax parser for parsing syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generating an object on the basis of GUI of the terminal.

10. (Original) The system of claim 9, wherein the token table comprises:

tokens defined in an XML document;

keywords defined in DTD for all documents provided to the handheld terminal; and

a list of elements which can be supported by each terminal.

11. (Original) The system of claim 9, wherein the word parser comprises:

a comment parser for recognizing a comment or a space and generating a token;

a markup start parser for recognizing a markup start tag and generating a token;

an attribute parser for parsing an attribute and generating a token; and

a parsed character data analyzer for analyzing parsed character data and generating a token.

12. (Original) The system of claim 9, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

13. (Original) The system of claim 9, wherein the contents model means a hierarchy of elements and an attribute list, and is defined in DTD for all documents provided to the handheld terminal.

14. (Original) The system of claim 9, wherein the syntax parser comprises:

an XML verifier for verifying whether a web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator for matching the parsed syntax and a GUI of the terminal.

15. (Original) A system for parsing web-document based on elements, comprising:

a token table comprising tokens defined in an XML document, keywords defined in DTD for all documents provided to the handheld terminal, and a list of elements, which can be supported by each terminal;

a word parser for extracting and separating all tokens of the document supplied to the terminal regardless of kind of a markup language used to compose the web-document by referring to a token table;

a contents model defined in DTD for all documents provided to the terminal and meaning a hierarchy of elements and an attribute list; and

a syntax parser for parsing syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generating an object on the basis of GUI of the terminal through the parsed syntax.

16. (Original) The system of claim 15, the word parser comprises:

a comment parser for recognizing a comment or a space and generating a token;

a markup start parser for recognizing a markup start tag and generating a token;

an attribute parser for parsing an attribute and generating a token; and

a parsed character data analyzer for analyzing parsed character data and generating a token.

17. (Original) The system of claim 15, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

18. (Original) The system of claim 15, wherein the syntax parser comprises:

an XML verifier for verifying whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator for matching the parsed syntax and a GUI of the terminal.

19. (Original) A handheld terminal comprising:

an integral parser for parsing a web-document composed of a predetermined markup language supplied from a web-server;

a memory for storing information parsed by the integral parser; and

an application program using information extracted from the integral parser.

20. (Original) A handheld terminal comprising an antenna, a CPU, a peripheral circuit, a vocoder, a memory and an audio codec, wherein the memory comprising:

an integral parser for calling a web-document supplied from a web-server regardless of a markup language used to compose the web-document and parsing the web-document on the basis of elements; and

an application program using information extracted from the integral parser.

21. (Currently Amended) The handheld terminal of claim 19~~ or 20~~, wherein the integral parser comprises:

a token table comprising tokens defined in an XML document, keywords defined in DTD for all documents provided to the handheld terminal, and a list of elements, which can be supported by each of the handheld terminals;

a word parser for extracting and separating all tokens of the document supplied to the terminal regardless of kind of a markup language used to compose the web-document by referring to a token table;

a contents model defined in DTD for all documents provided to the terminal and meaning a hierarchy of the elements and an attribute list; and

a syntax parser for parsing syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generating an object on the basis of GUI of the terminal through the parsed syntax.

22. (Original) The system of claim 21, the word parser comprises:

a comment parser for recognizing a comment or a space and generating a token;

a markup start parser for recognizing a markup start tag and generating a token;

an attribute parser for parsing an attribute and generating a token; and

a parsed character data analyzer for analyzing parsed character data and generating a token.

23. (Original) The system of claim 21, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

24. (Original) The system of claim 21, wherein the syntax parser comprises:

an XML verifier for verifying whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator for matching the parsed syntax and a GUI of the terminal.

25. (Currently Amended) The handheld terminal of claim 19~~ or 20~~, wherein the application program comprises an object based on a GUI of the handheld terminal.

26. (Original) A method for parsing a web-document supplied from a web-server, the web-document being composed of a predetermined markup language, the method comprising the steps of:

(a) reading a token from the web-document by referring to a token table, extracting and separating the token;

(b) if the extracted and separated token is not a defined start tag or if the token is a comment or a space, ignoring the token;

(c) when the extracted and separated token is recognized as the defined start tag, parsing an attribute of an element from the token and storing GUI-related information of the element;

(d) parsing contents of the element after parsing the attribute of the element;

(e) as the result of the step (d), if the contents of the element are parsed character data,

storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, determining whether an end tag appears;

(f) as the result of the step (e), if the end tag does not appear, repeating from the step (a), and if the end tag appears, determining whether the end tag corresponds to the defined start tag; and

(h) as the result of the step (f), if the end tag corresponds to the defined start tag, terminating, and otherwise, ignoring and returning.

27. (Original) The method of claim 26, wherein the step (c) comprises the steps of:

if the extracted and separated token does not include a defined start tag, reading the data continuously until the end tag appears, thereby ignoring the token; and

reading a new token.